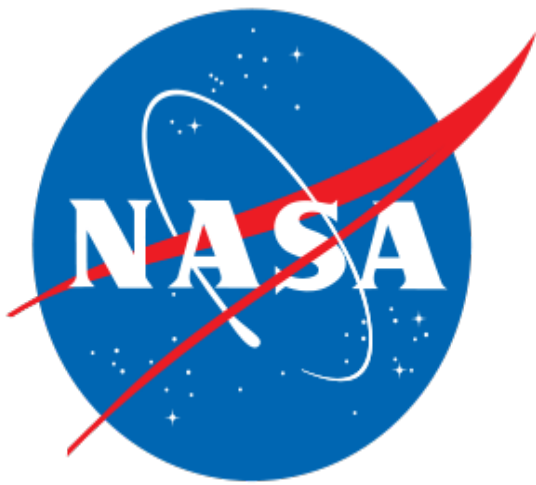


“Integrated Graphics Operations and Analysis Lab Development of Advanced Computer Graphics Algorithms”



NASA Lyndon B. Johnson Space Center, Houston, TX
Engineering Directorate (EA)

Ira M. Wheaton Jr.
MUST Program
Mentor: Sharon Goza
May 31, 2011 – August 5, 2011

Introduction

The Integrated Graphics Operations and Analysis Laboratory (IGOAL) is an important section of NASA, as the group “supports engineering visualization throughout the Johnson Space Center, additional NASA centers, and other institutions. IGOAL Engineering Visualization broadcasts quality computer generated video with near-photo realistic rendering.” This team uses transformative and cutting-edge technology that displays proof of concept and demonstrates operational scenarios. Moreover, one of the most rewarding functions of the IGOAL is its dedication to NASA’s commitment to education and the next generation of American scientists and engineers. The IGOAL visualizations express NASA’s missions to people of all ages in the modern, effective way of using technology. Software is distributed to schools at no cost to them, and it includes training and support to bring results of scientific data to the classroom. These are just a few reasons why the IGOAL is essential to the future of NASA.

Abstract

The focus of this project is to aid the IGOAL in researching and implementing algorithms for advanced computer graphics. First, this project focused on porting the current International Space Station (ISS) Xbox experience to the web. Previously, the ISS interior fly-around education and outreach experience only ran on an Xbox 360. One of the desires was to take this experience and make it into something that can be put on NASA’s educational site for anyone to be able to access. The current code works in the Unity game engine which does have cross platform capability but is not 100% compatible. The tasks for an intern to complete this portion consisted of gaining familiarity with Unity and the current ISS Xbox code, porting the Xbox code to the web as is, and modifying the code to work well as a web application. In addition, a procedurally generated cloud algorithm will be developed. Currently, the clouds used in AGEA

animations and the Xbox experiences are a texture map. The desire is to create a procedurally generated cloud algorithm to provide dynamically generated clouds for both AGEA animations and the Xbox experiences. This task consists of gaining familiarity with AGEA and the plug-in interface, developing the algorithm, creating an AGEA plug-in to implement the algorithm inside AGEA, and creating a Unity script to implement the algorithm for the Xbox. This portion of the project was unable to be completed in the time frame of the internship; however, the IGOAL will continue to work on it in the future.

Goals and Purposes

This array of assignments not only provides an intern with a broad exposure to integrated graphics analysis, but also helps support engineering visualization throughout the Johnson Space Center, additional NASA centers, and other institutions. Additionally, the intern will be given the opportunity to assist IGOAL with continuing to contribute to NASA's educational initiative.

ISS XBOX Experience Project

This project focuses on porting the ISS fly-around experience to the web. Prior to my summer arrival, the IGOAL created a mock experience of the International Space Station, using the Unity Game Engine. The first-person game experience was designed to be compatible with the XBOX 360 Kinect motion sensing input device, where players can physically navigate through the Space Shuttle by performing a series of body movements. Players basically float through the different modules of the ISS, and each module can take the player to a different scene where he or she can play a game or attempt to complete a mission. Some examples are the following: the player can enter the gravity experience where they attempt to accumulate points by throwing tennis balls at various targets while subject to the gravity of different planets, the player can enter the TrainR2 experience where he or she performs different arm poses in an

attempt to have Robonaut 2 duplicate the same poses to accumulate points, and the player can enter the solar array repair mission where he or she attempts to repair a solar array within a certain period of time. Also, seven different informational balls are scattered throughout the modules. The “info balls” have valuable facts about the different modules for the player to read. All of this was put together to form a fun, unique integrated graphical experience of the International Space Station.

As previously stated, this experience was built to run on the XBOX 360 Kinect, and my assignment was to take this experience and make it available to access on the web. This required many different steps. First, the Unity Web Player plug-in had to be downloaded. Ideally, we could have simply used Unity’s cross platform capability to simply build and run the project using the Web Player. However, much of the Unity scripts and graphics were catered to an XBOX 360 Kinect environment, making the switch less than 100% compatible. This caused the task to run into many obstacles that would eventually be conquered.

After gaining familiarity with Unity and the ISS XBOX experience code, I realized that much of the code actually catered to the web platform, which would still enable a user to navigate through the Space Shuttle and ISS using a mouse and keyboard. Nonetheless, one of the more obvious problems that we faced was the graphics in the game. Most of the graphics were expectedly catered to XBOX 360 Kinect movements. For example, in order to brake or slow down while using the Kinect, players must perform a “T-pose” where they place their arms horizontal to their shoulders, and a texture in the corner of the screen serves as a reminder by showing a stick figure performing the “T-pose”. Also, some of the directions to enter or leave different games/missions tell the user to perform certain body movements, which clearly would

not be the case for a Web Player user. As a result, I used Microsoft Paint and Adobe Photoshop to modify these graphics and replace them in the Unity project file.

Next, another issue that we came across was the overall compatibility of the games and missions with a PC. Once again, these extracurricular segments are heavily dependent on the XBOX 360 Kinect's ability to recognize real world three dimensional motions. The solar array repair mission requires one to use his or her hands to reach out and grab tools to use for the repair. Similarly, the R2 poses scene requires the user to form poses using his or her arms for Robonaut to duplicate, and the gravity experience requires the player to perform a throwing motion with their arms to simulate throwing the tennis balls. All of these actions were impossible for a device to recognize without the Kinect technology. Therefore, we decided to simply eliminate these game scenes from the Web Player Build and remove many of the scene components in the Unity file to create a more concise appearance.

The last major obstacle that we experienced was compatibility with the info ball timers. Once the user activates an info ball, he or she then has 5 minutes to find, activate, and read the remaining 6 info balls. As each info ball is activated, a small pie chart timer is drawn to show the user how much time remains. However, once we built the scene to the Web Player, we noticed that the pie chart timers were not being drawn. The previous code that drew the timers consisted of different functions from the GL low-level graphics library, but this library was not compatible with the Unity Web Player. As a result, I had to search for, develop, and implement an alternate algorithm that would draw the pie chart style timers. This was creating by using different textures including a left and right half of a circle, a full circle of a different color, and a right half circle of the same color as the full circle. First, we draw the full circle using the `GUI.DrawTexture` function. Next, we draw the left and right half circles on top of the full circle.

Once the timer is activated, we then simply rotate the right half circle pieces at a certain rate by using the `GUIUtility.RotateAroundPivot` helper function to rotate a texture around a point. The right piece will rotate behind the left piece. Once past the halfway mark, we then draw the right half circle, which has the same color as the full circle, to block the user from noticing that the left half circle is now rotating behind it. Once the time is complete, the full circle will show, and the user must start again. This method was implemented and worked well with Unity Web Player. Now this project is in the process of being uploaded to NASA's educational website for all to experience.

The importance of this project is that it gives people of all ages a change to virtually experience the International Space Station, which works well with NASA's desire to educate and inspire others to continue to explore. Additionally, we hope that it will also inspire more students to study and pursue careers in the Science, Technology, Engineering, and Mathematics fields, which is another part of NASA's education mission.

Procedurally Generated Cloud Algorithm

This project focuses on developing an algorithm to generate clouds procedurally. Currently, the ISS unity web experience simply uses a texture that simulates the Earth from the view of the ISS. The texture is only one octant of the Earth, and it simply rotates to create an illusion. The desire is to create dynamically generated clouds that appear more authentic. Before my arrival, the IGOAL developed a large, complex algorithm as an attempt to accomplish the goal. For a few weeks, I gained familiarity with the code and attempted to find better ways to polish the code and create a better algorithm. Unfortunately, I was unable to complete this portion of the project. However, the IGOAL will continue to work on this feat after my departure.

Impact on Career Goals

This was my second MUST Internship at the NASA Johnson Space Center, and yet again it proved to be another valuable experience that has inspired me to pursue my career goals of receiving my Ph.D. in Mathematics, completing research, and teaching mathematics. In the Fall 2011 Semester, I will be enrolling in the Financial Mathematics Ph.D. Program at The Florida State University. To be successful in this program requires a strong mathematical aptitude, analytical skills, research skills, and extensive computer programming experience. Both of my internship experiences at the NASA Johnson Space Center have developed me in these areas. Particularly, this summer internship has truly enhanced my analytical skills and computer programming experience. My knowledge of the C++ programming language, as well as C# and JavaScript, has increased, which is extremely beneficial because understanding these languages is vital to succeed in the Financial Mathematics field of study. Furthermore, learning to work individually and in groups is an essential part of succeeding in graduate school. These internship experiences have groomed me into a curious, diligent worker. I have learned how to effectively work with others and to do individual research while still under supervision, as I learned to search for answers to certain questions. Furthermore, I learned to not be afraid to communicate my ideas and/or listen to the instructions of a supervisor. This quality of being independent, but teachable, is one that is appreciated by most mentors and faculty members of graduate research programs.

Professional mentorship is one of the many pillars of the MUST Program. I can testify that I received excellent mentorship throughout my internship experience. My mentors, both formal and informal, were highly instrumental in communicating and breaking down complex concepts that are typically difficult for most people to comprehend. They were very

understanding and always willing to help me with any problem that I had. Moreover, my mentors made me feel comfortable from the start of my internship, as they treated me as though I was a part of the team. They also made sure that I had a complete experience, whereas I would not only do plenty of work, but also have fun and enjoy my time there. They helped to schedule different tours of the facility to see much of the cool things that are done at the Johnson Space Center, in Houston, TX, and anything involving NASA worldwide. I believe that this was the ex-factor in me truly enjoying my internship experience. The excellent mentorship that surrounded me throughout this internship justly influenced my mindset and productivity. I hope to receive this kind of mentorship throughout my future research and career experiences.

Figures, Graphs, and Tables

Figure 1: A screen shot of the start-up view of the ISS Experience inside of the Unity Game Engine



Figure 2: A screen shot of the Gravity Experience game inside of the Unity Game Engine



Figure 3: A screen shot of an “info ball” inside of the Unity Game Engine

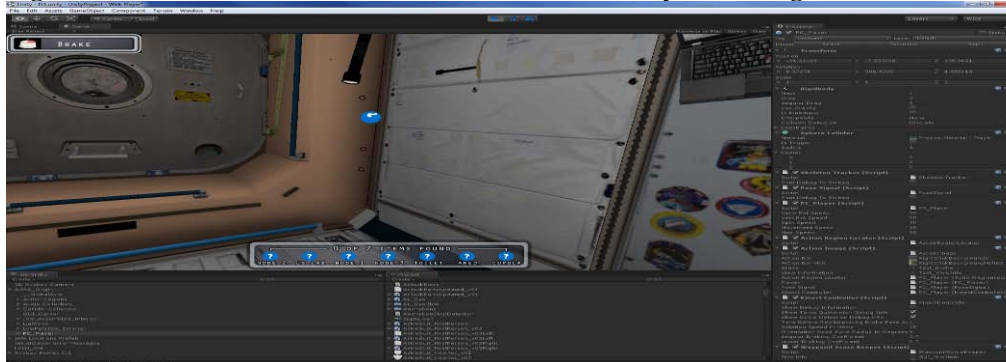


Figure 4: A screen shot of the revised “info ball” timers

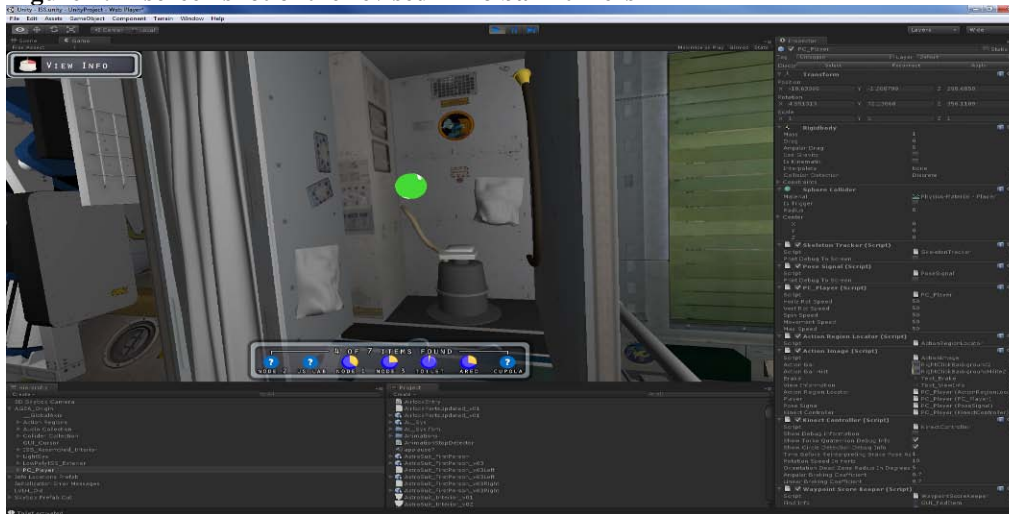


Figure 5: A screen shot of the information displayed by the Node 1 “info ball”

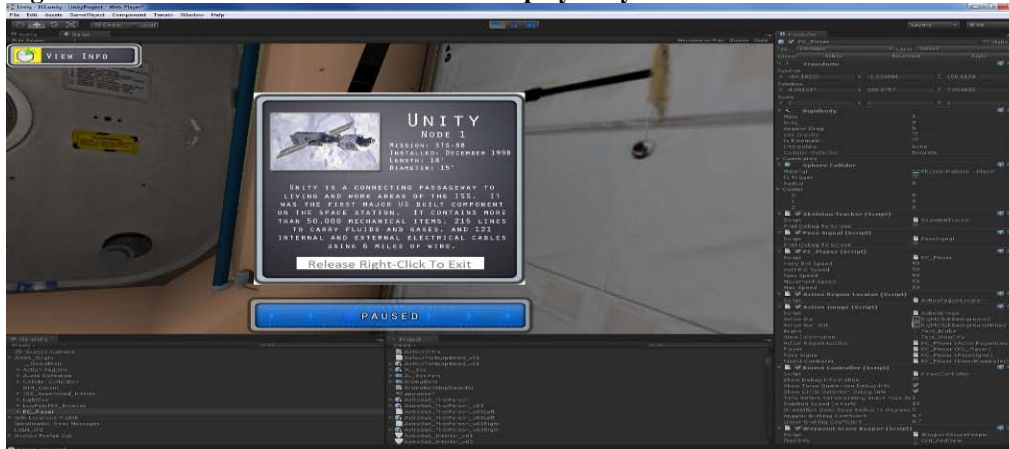


Figure 6: A screen shot of the view of Earth from the Cupola Window

